Amendment to Specification

Please amend the paragraph on page 7, lines 5-27 as follows:

Fig. 1 illustrates a port interface (PIF) circuit 110 including a pipelined multi-tasking processor (microcontroller) 160. Port interface 110 includes four full-duplex ports that provide an interface between ATM switch 120 and respective four Ethernet segments (not shown) each of which is connected to a corresponding MAC 130.0-130.3. In each port "x" (x=0,1,2,3) the data between the Ethernet segment and the ATM switch 120 flows through a corresponding MAC 130.x and a corresponding slicer 140.x. The slicer performs the well-known ATM SAR function, segmenting the Ethernet frame into ATM cells and appending ATM headers to the cells on the way to ATM, and assembling the frame from the cells on the way to the Ethernet. In some embodiments, the ATM switch interface to PIF 110 operates in frame mode in which the ATM switch transmits a frame of cells to a slicer 140 with no intervening cells. Slicers 140 use the AAL-5 protocol. The frame mode is described, for example, in U.S. patent application 08/706,104 "Cell Queuing in ATM Switches" filed August 30, 1996 by A. Joffe et al., now U.S. patent no. 6,128,278. See also PCT application PCT/US97/14821 filed August 28, 1997 and incorporated herein by reference.

Please amend the paragraph on page 8, lines 3-13 as follows:

In addition to performing protocol transformations (e.g. ATM/Ethernet transformations), PIF 110 can perform IP routing, layer-2 switching, or other processing as determined by the software executed by the PIF microcontroller 160. See the description below in connection with Figs. 3A, 3B. See also U.S. patent application no. 09/055,044 "SYSTEMS AND METHODS FOR DATA TRANSFORMATION AND TRANSFER IN NETWORKS" (now U.S. patent no. 6,307,860), attorney docket number M-4855 US, filed by A. Joffe et al. on April 3, 1998 ~~the same date as the present application~~ and incorporated herein by reference.

Please amend the paragraph on page 8, lines 19-30 as follows:

In Fig. 1, the data flow between each slicer 140.x and the corresponding MAC 130.x is controlled by a corresponding channel 150.x (also called channel "x" below, i.e. channel 0, 1, 2 or 3). The channels 150 executes commands from microcontroller 160. In some embodiments, the four channels 150.x are implemented by a single channel circuit that performs the function of the four channels 150 using time division multiplexing. See the aforementioned U.S. patent application no. 09/055,044 ~~attorney docket number M-4855-US~~, "SYSTEMS AND METHODS FOR DATA TRANSFORMATION AND TRANSFER IN NETWORKS" incorporated herein by reference.

Please amend the paragraph starting on page 10, line 25 and ending on page 11, line 7 as follows:

(1) The corresponding input control block 210 (i.e. 210I or 210E) stores the incoming data in the corresponding data FIFO 220. When a sufficient portion of a data frame has been received to enable the microcontroller to start address translation or other processing (e.g., when the IP address and hop count have been received in IP routing embodiments), input control 210 writes a request to respective request FIFO 230. The number of frame bytes received before the request is written to FIFO 230 is defined by microcontroller-writable registers as described in the aforementioned US patent application no. 09/055,044 ~~attorney docket number M-4855-US~~.

Please amend the paragraph on page 15, lines 15-24 as follows:

At one or more of sub-stages 290DA.3, 290IP.3, 290SA.3, the microcontroller writes commands to the command FIFO 260I for the respective data flow (i.e. respective sub-channel). The commands may instruct the channel 150 to drop the frame, or to forward the frame to respective slicer 140. If the frame is forwarded, the channel may supply the VPI/VCI to the slicer and, possibly, increment the ~~to~~ IP hop count and/or replace the source address with the address of respective MAC 130, as directed by the commands.

Please amend the paragraph on page 17, lines 3-20 as follows:

In some embodiments, having a single task for each ingress flow and each egress flow does not fully load the microcontroller, and therefore more than one task for each half-duplex data flow are provided to enable the microcontroller to process more than one frame in each data flow in parallel. This is illustrated by the following considerations. The demands on the [[of]] microcontroller speed are the greatest when the Ethernet frames are short, because the same processing of Figs. 3A, 3B has to be performed both for short and long frames. The shortest Ethernet frame has 64 bytes. Suppose for example that the four Ethernet ports are 100 MB/sec ports and the ATM ports are 155 MB/sec. At 100 MB/sec, the shortest frame goes through the Ethernet port in 5.12 microseconds. Therefore, the microcontroller and the search machine have to process the frame in 5.12 + 1.6 = 6.72 microseconds (1.6 microseconds is the interframe gap).

Please amend the paragraph on page 30, lines 10-11 as follows:

State S5 is similar to S4, but describes ~~describe~~ Egress Task 1 writing and owning the command FIFO.

Please amend the paragraph starting on page 33, line 27 and ending on page 34, line 5 as follows:

(2) The channel 150 writes the semaphore register. This is indicated by cstrobe being asserted (Table A4-1 in Addendum 4) and csem[5] being at 1. The channel accesses the semaphore register to send an indication to microcontroller 160 when commanded by a channel command. See the aforementioned U.S. patent application no. 09/055,044 ~~attorney docket No. M-4855-US~~ "Systems and Methods for Data Transformation and Transfer in Networks" incorporated herein by reference.

Please amend the last paragraph on page 52 as follows:

This area is described in the aforementioned U.S. patent application no. 09/055,044 ~~attorney docket number M-4855-US~~.

Please amend the paragraph on page 53, lines 12-19 as follows:

Each register r0.0-r7.7 is 1 byte wide. 8 consecutive bytes can be read in parallel from the register file. To form a 7-bit ~~register~~ address of an 8-byte register word, the register number (0 through 63) is concatenated with the bank ID which itself is a concatenation of the channel ID "CHID" and the task pair number SN (0 or 1); the address MSB is 0 to indicate register file 312 (versus special registers 314).

Please amend the paragraph on page 55, lines 9-11 as follows:

The resources and the data memory 316 (note types b and c) ~~(note types (b) and (c))~~ are mapped into the special registers to simplify their access.